

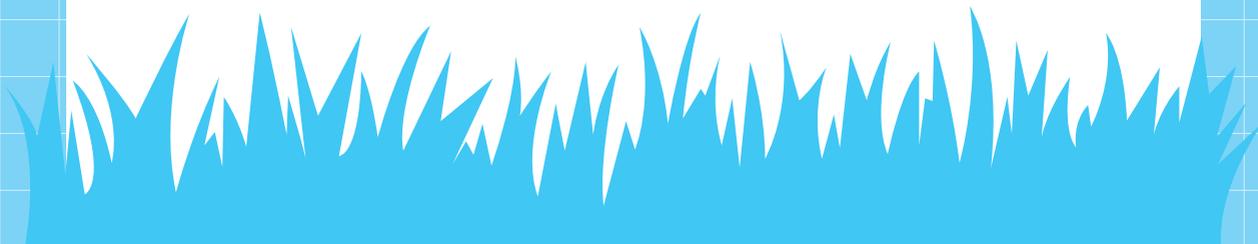


项目 17 神奇的按钮



按钮具有按下（高电平）和抬起（低电平）两种状态，默认状态为抬起。生活中的按钮可以说无处不在，遥控器、计算器、手机、计算机等各种电子设备上的按键，它们的功能与按钮一样。

本项目学习按钮模块的使用方法，编写代码通过按钮控制 LED 灯。通过学习，掌握定义引脚为电平输入的方法和读取输入信号的方法，使用循环结构编写代码实现控制要求。



任务 17.1 简单的按钮

连接按钮模块，编写代码，实现简单的按钮功能：按下按钮，LED 灯亮；松开按钮，LED 灯灭。

使用按钮控制 LED 灯的控制逻辑是，通过按钮输入的电平信号，经主板分析处理，再通过 LED 输出。主板分析处理的过程就是代码执行的过程。

1. 硬件及其连接

一个电子产品的开发过程要经历产品硬件开发和软件开发，硬件开发要进行电路原理图设计和 PCB 板图设计→PCB 专业厂家制作 PCB→焊接电子元器件→调试产品→产品使用→小批量生产→小范围试用→产品定型→产品质检→生产许可→大批量生产→投放市场→售后服务。

学习时，硬件制作要花费一定费用，为了节约费用一般自己制作 PCB（相关制作方法这里不详述，读者可自行查阅资料），也可以用面包板和万能板自己搭建。万能板正、反两面如图 17-1 所示，器件可以直接焊接在板上，这里不详述。

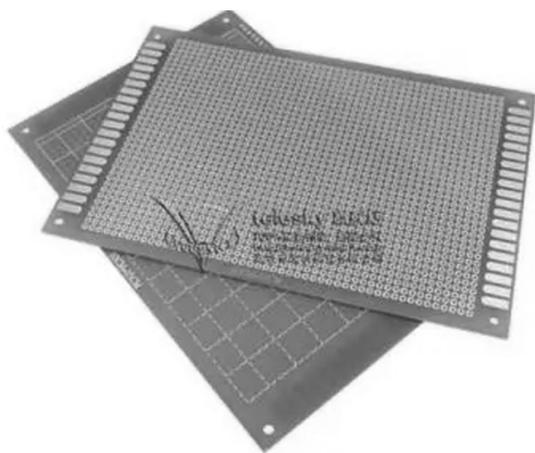


图 17-1 万能板正、反两面

面包板使用接插件即可实现电路搭建，无须焊接，因此更加安全、便捷，

缺点是有接触不良的现象，要注意检查和接插牢靠。下面具体介绍面包板及其使用。

(1) 面包板。面包板的种类很多，结构和作用不尽相同。如图 17-2 所示的面包板外观，左右两侧标有 - 和 + 符号的为电源插孔，中间部分标有字母和数字符号的是接线端子插孔。

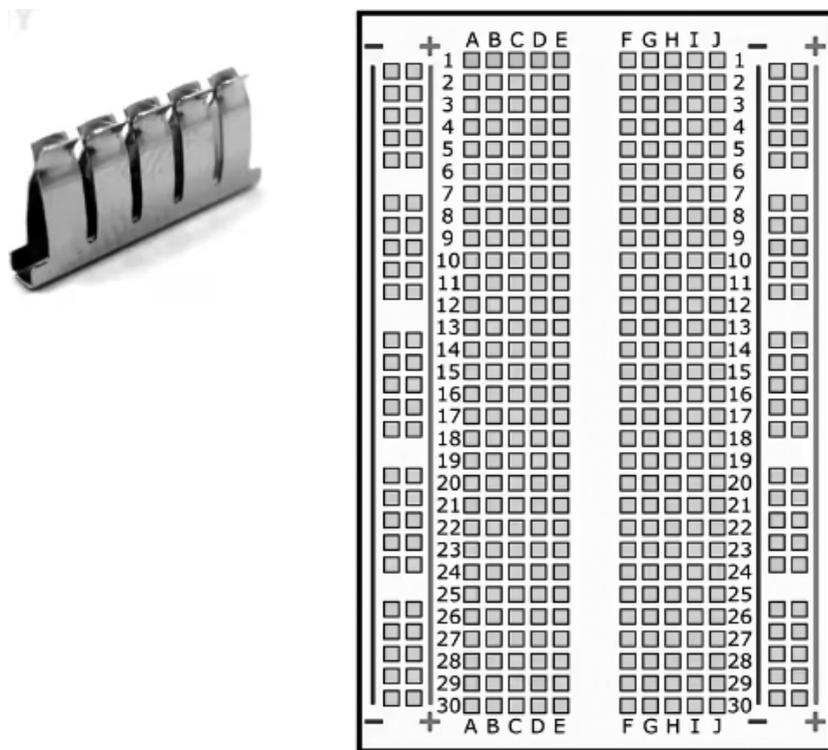


图 17-2 面包板外观

只了解外观还不够，想要了解每个插孔的使用方法，还需要了解面包板的内部结构。如图 17-3 所示，黄色线条连接起来的插孔，代表它们是连通的一组。面包板的每一个接线插孔其实是一个个金属夹子，并且是 5 个一组排列的。如 A1、B1、C1、D1、E1 这 5 个插孔是一组，电路上是相通的。而 A1 孔不与 A2、B2、C2、D2、E2 相连，也不与 F1、G1、H1、I1、J1 相连。

(2) 搭建电路。选择主板上引脚 3 为按钮信号输入，引脚 10 为 LED 信号输出。这样，就可以正确使用面包板搭建电路了，读者可以自行搭建。本项目不使用面包板，而使用小集成模块。

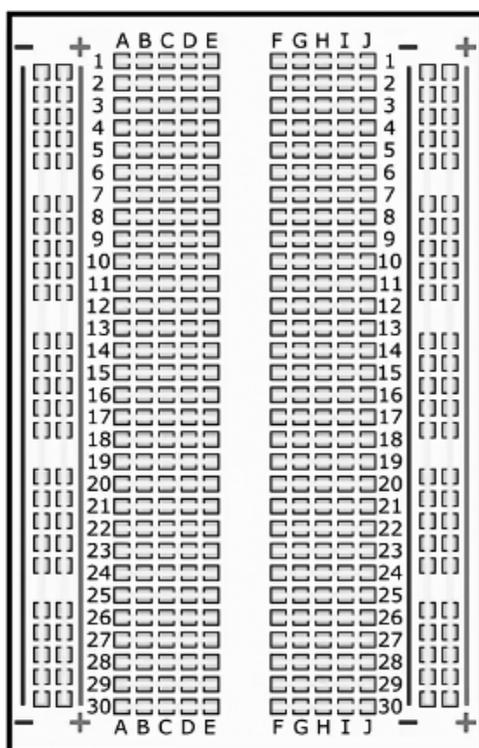


图 17-3 面包板的内部结构

图 17-4 中连接在引脚 3 上的小模块为按键，连接在引脚 10 上的小模块

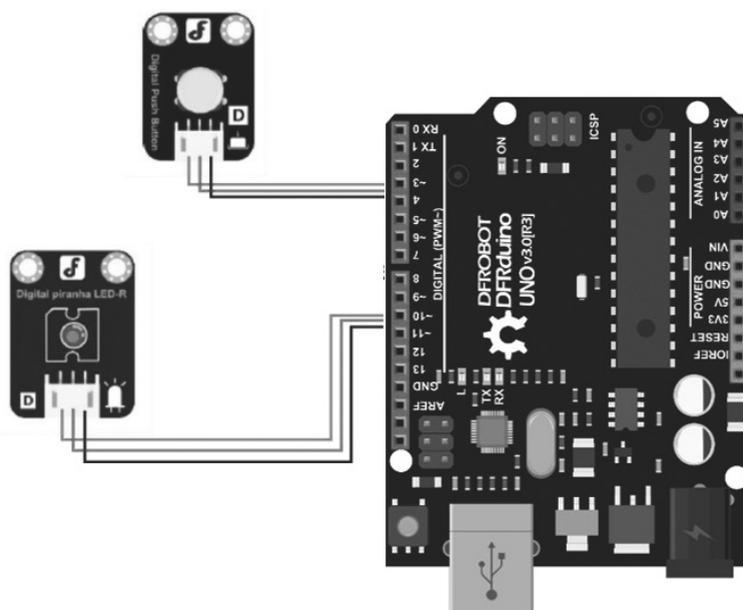


图 17-4 硬件接线图





Python 编程入门 (下)

为指示灯, 将小模块摆放整齐, 一一核对避免出错。各小模块有 3 个引脚: 一个引脚接地 (黑色线), 一个引脚接电源正极 (红色线), 另一个引脚接信号 D (绿色线), 小模块上都有标注。将按键小模块的信号线 (绿色线) 接主板端口 3, 指示灯小模块信号线 (绿色线) 接主板引脚 10, 接好各小模块的电源线, 如图 17-4 所示。

2. 程序设计思路

本项目是将输入输出通过主板 CPU 联系在一起的项目, 这是智能控制的最基本的设计思想, 一个智能控制系统要有输入输出设备, 还要有大脑 (CPU), 又称为控制设备, 如图 17-5 所示。编程时先要判断输入状态, 按照输入状态的情况, 经过 CPU 处理后, 再控制输出动作。

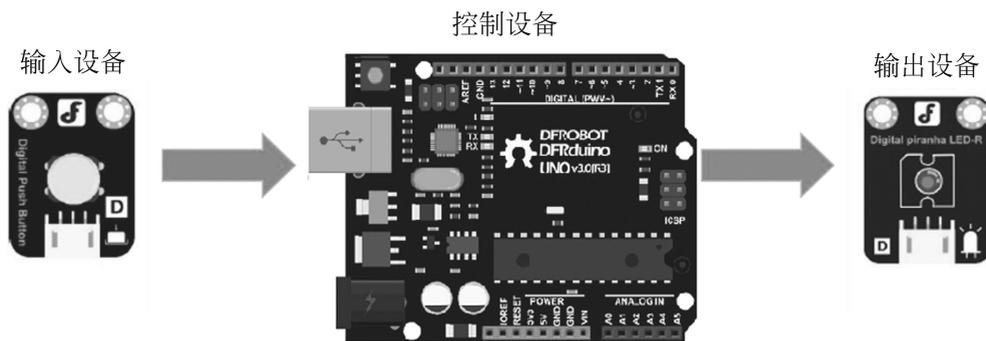


图 17-5 智能控制系统

3. 代码编写

完成硬件接线, 就可以编写代码了。打开 Mind+ 软件, 按之前学习的方法新建文件, 并命名为“简单的按钮”。

本次任务没有延时, 不需要导入时间库。LED 灯控制程序参照项目 15 中代码编写的方法编写, 按照本次任务要求, 编写按钮控制程序如图 17-6 所示。

注意: (1) 初始化中, 电平输入和电平输出的书写区别。

(2) 检测按钮状态是读取输入引脚的电平, 指令是 `read_digital`。

逐行检查代码, 确认无误后, 连接通信线上传至主板, 单击“运行”按钮,

```

简单的按钮.py x
1  from pinpong.board import Board,Pin  #导入主板和pinpong库
2  Board("uno").begin()                #主板初始化
3  anniu=Pin(Pin.D3,Pin.IN)             #初始化, 引脚3为电平输入
4  led=Pin(Pin.D10,Pin.OUT)             #引脚10为电平输出
5
6  while True:                          #循环开始
7      if(anniu.read_digital()==1):     #如果检测到按钮按下
8          led.write_digital(1)         #高电平输出, LED灯亮
9      if(anniu.read_digital()==0):     #如果检测到按钮抬起
10         led.write_digital(0)         #输出低电平, LED灯灭

```

图 17-6 编写按钮控制程序

观察运行结果。如果代码没有错误,可以发现,没有任何操作时,LED灯不亮,相当于按钮抬起时的状态。当按下按钮时,LED灯亮,松开时LED灯灭,实现了简单的按钮控制。

4. 保存项目

完成后,选择“项目”→“保存项目”命令,项目名称为“神奇的按钮”,选择保存位置,单击“保存”按钮即可。

下次需要使用该项目时可以直接打开已经保存的项目,方便查看和修改。

任务 17.2 延时灯

任务 17.1 实现了当按钮按下,灯亮;按钮松开,灯马上熄灭。对于有些控制场合,需要灯延长一段时间再灭,如楼道灯。下面制作一种延时灯,按下按钮,灯亮,松开按钮时,等待一段时间后灯自动熄灭。

1. 任务描述

按下按钮灯亮,松开按钮 3 秒后灯灭。

2. 任务分析

硬件接线与任务 17.1 相同,不改变引脚号。硬件和接线方法参照图 17-4。

3. 程序设计思路

最简便的方法是在任务 17.1 的基础上修改程序，在循环语句中增加延时，只需要一个判断语句就可以了。这里需要导入时间库。

4. 编写代码

按照上述思路，编写代码。打开项目“神奇的按钮”，按之前所学方法新建文件，命名为“延时灯”。复制之前的代码，再修改，代码参考如图 17-7 所示。

```
延时灯.py x
1  from pinpong.board import Board,Pin  #导入主板和pinpong库
2  import time                          #导入时间库
3  Board("uno").begin()                 #主板初始化
4  anniu=Pin(Pin.D3,Pin.IN)             #初始化, 引脚3为电平输入
5  led=Pin(Pin.D10,Pin.OUT)            #引脚10为电平输出
6  while True:
7      if(anniu.read_digital() == 1):  #如果检测到按钮按下
8          led.write_digital(1)        #高电平输出, LED灯亮
9          time.sleep(3)               #延时3秒
10         led.write_digital(0)        #低电平输出, LED灯灭
```

图 17-7 “延时灯” 代码

连接通信电缆，单击“运行”按钮，观察程序运行结果。此时，按下按钮，LED 灯亮，松开后，LED 灯继续保持亮的状态，3 秒后自动熄灭。实现了延时灯的效果。

任务 17.3 扩展阅读：电平信号

常见的电平标准有 TTL、COMS、LVDS、ECL、RS232、RS485 等，其中包含切换速度较快的 LVDS、GTL、CMOS，每种电平的供电电压和电平标准不一样。

TTL (Transistor-Transisitor Logic, 晶体管 - 晶体管逻辑) 电平信号使用

得最多，通常数据表示采用二进制，+5V 等价于逻辑 1，0V 等价于逻辑 0，这被称作 TTL 信号系统，是计算机处理器控制的设备内部各部分之间通信的标准技术。

在数字电路中，由 TTL 电子元器件组成的电路就是 TTL 电路。电路的第一代是电子管，第二代是晶体管，这段时期，集成电路发展迅猛，各大厂商竞相出台自己的电平标准，导致各芯片之间的连接出现一些接口电路，这也导致额外的浪费，后来各大生产商统一了晶体管的集成电路的端口电平的范围，这就是 TTL 电平。

电平是个电压范围，规定输出高电平大于 2.4V，输出低电平小于 0.4V。在室温下，一般输出高电平是 3.5V，输出低电平是 0.2V。最小输入高电平是 2.0V，最大输入低电平是 0.8V，噪声容限是 0.4V。

任务 17.4 总结与评价

(1) 展示作品，交流自己学习中的发现、疑问和收获。

(2) 思考：

① 更换不同的引脚，编写程序实现按钮功能。

② 按钮控制 LED 灯有多种方式，除了本项目描述的，还有其他方式。

按下按钮，灯亮并保持，再按下按钮灯灭并保持，尝试编写程序实现这样的按钮控制方式。

(3) 项目 17 已完成，在表 17-1 中画☆，最多画 3 个☆。

表 17-1 项目 17 评价表

评价描述	评价结果
能编写任务 17.1 中的代码，并测试成功	
能编写任务 17.2 中的代码，并测试成功	
经过检查和校对，能主动发现编写错误，并修正	
能对原程序做适当修改，并测试成功	